

1. A method of programmatically selecting system designs from a system design space, the method comprising:

5 specifying system designs as combinations of component designs
from respective component design spaces;
applying component quality filters to the component design spaces
to produce component quality sets of designs; and
forming a Cartesian product of the component quality sets to
10 obtain a set of system designs.

2. The method of claim 1, further comprising applying component validity filters to respective component design spaces before applying the component quality filters, wherein the component quality sets of designs
15 include only designs satisfying respective component validity filters.

3. The method of claim 1, further comprising applying a system validity filter to the set of system designs to produce a validity filtered set of system designs.

4. The method of claim 3, further comprising applying a system quality filter to the set of system designs.

5. The method of claim 1, further comprising applying a system
25 quality filter to the set of system designs.

6. A method of programmatically selecting system designs that are specified by combinations of component designs, the method comprising:

preparing component validity sets for each of the component designs by applying component validity filters to corresponding component design spaces, the component validity filters defined by corresponding component validity predicates; and

5 forming a set of system designs that is a Cartesian product of the
 component validity sets.

7. The method of claim 6, wherein the component designs are specified by component parameters, and the component validity filter for each component is independent of the component parameters of other components.

8. The method of claim 6, further comprising applying a system validity filter to the Cartesian product of the component validity sets.

9. The method of claim 6, further comprising applying a system quality filter to the Cartesian product of the component validity sets.

10. The method of claim 6, further comprising applying a system
20 evaluation function and a system quality filter to the Cartesian product of
the component validity sets after applying a system validity filter.

11. The method of claim 10, further comprising applying a component evaluation function and a component quality filter to the component validity sets.

12. The method of claim 6, further comprising applying a component evaluation function and a component quality filter to at least one of the component validity sets before forming the set of system designs.

5

obtaining a lower bound for an evaluation metric for a system design, wherein the system design includes the partial system design; and

10

15

preparing component quality sets by applying corresponding component evaluation functions and component quality filters to the component validity sets; and

25

15. The method of claim 14, further comprising applying a system validity filter to the Cartesian product of the component quality sets.

30

17. The method of claim 16, wherein the component evaluation functions and the system evaluation function produce component evaluation metrics and system evaluation metrics, respectively, and the system evaluation metrics are obtained from the component evaluation metrics.

18. A computer readable medium comprising computer executable instructions for performing the method of claim 1.

19. A computer readable medium comprising computer executable instructions for performing the method of claim 6.

20. A computer readable medium comprising computer executable instructions for performing the method of claim 14.

21. A method of programmatically selecting a system design from a set of system designs, comprising:

defining a system validity predicate that is a function of two or more terms;

defining partial validity predicates by expressing the system validity predicate in a canonical form;

applying partial validity filters that are defined by the partial validity predicates to the system designs to obtain partial validity sets; and

combining the designs from the partial validity sets to obtain sets of designs satisfying each of the two or more terms.

22. The method of claim 21, where each of the partial validity predicates is in product form.

23. The method of claim 21, wherein the partial validity predicates are mutually exclusive.

24. A method of programmatically selecting a set of system
5 designs, comprising:

selecting a system validity filter defined by a system validity predicate, the system validity predicate including one or more partial validity predicates that define partial validity filters;

applying the partial validity filters to the system designs;

10 forming partial validity sets that include system designs satisfying
 respective partial validity filters;

applying an evaluation function to the system designs of the partial validity sets, the evaluation function producing an evaluation metric for each system design;

15 applying a quality filter to the system designs of the partial validity
sets, the quality filter comparing and selecting system designs based on
the evaluation metrics and producing respective partial quality sets; and
combining the partial quality sets to form a first quality set.

20 25. The method of claim 24, further comprising applying the
quality filter to the first quality set.

26. The method of claim 24, wherein each of the partial validity predicates is in product form.

25

27. The method of claim 26, wherein the system validity predicate is a product of the partial validity predicates.

28. The method of claim 26, wherein the partial validity sets are
30 combined to form two or more system validity sets.

29. A computer readable medium having computer executable instructions for performing the method of claim 24.

5 30. A computer readable medium having software for performing
the method of claim 25.

31. A method of programmatically selecting a design for a cache memory, comprising:

10 selecting components for the cache memory;
 determining component Pareto sets for the components;
 preparing a combined Pareto set from the component Pareto sets;
 and
 selecting a cache memory design from the combined Pareto set.

15

32. A method of selecting a design for a processor system, the processor system including a processor and a cache memory, the method comprising:

20 preparing a component Pareto set for the processor;
 preparing a component Pareto set for a cache memory;
 preparing a combined Pareto set from the component Pareto sets
 of the processor and the cache memory; and
 selecting a processor system design from the combined Pareto
 set.

25

33. A method of programmatically generating a set of designs for a processor system, comprising:

dividing the processor system into at least a processor component and a memory component;

preparing component validity sets for the processor component and the memory component;

forming a Cartesian product of the component validity sets to produce a processor system validity set.

5

34. The method of claim 33, further comprising expressing the system validity function in a logical canonical form.

10

35. A method of designing a processor system that includes a processor component and a memory component, comprising:

determining component validity sets for the processor component and the memory component;

dividing at least one of the component validity sets into subsets;

15

and

generating sets of system designs by combining component designs from the component validity sets and the subsets.

·20

36. A method of generating a set of partial validity predicates for a system design that includes component designs for at least a first component and a second component, the method comprising:

obtaining a system validity function defined by a system validity predicate; and

25

identifying coupled terms in the system validity predicate, the coupled terms including parameters of the components.

37. The method of claim 36, wherein the system design is processor system design and the components include a processor component and a memory component.

30

38. The method of claim 37, further comprising expanding the coupled terms to obtain singleton terms containing parameters of only the processor component and singleton terms containing parameters of only the memory component.

5

39. The method of claim 36, further comprising expanding the coupled terms to obtain singleton terms containing parameters of only a first component and singleton terms containing parameters of only a second component.

10

40. The method of claim 39, further comprising expressing the system validity predicate in canonical form.

41. The method of claim 36, further comprising expressing the
15 system validity predicate in canonical form.

15

THE UNIVERSITY OF CHICAGO